Search in Multimedia Databases Using Similarity Distance

Mousa AL-Akhras
King Abdullah II School for Information Technology, The University of Jordan, Amman 11942, Jordan
mousa.akhras@ju.edu.jo

Mohammed Alawairdhi
College of Computer and Information Sciences, Al-Imam Muhammad Ibn Saud Islamic University, Saudi Arabia
mawairdhi@imamu.edu.sa

Dana Beidas
King Abdullah II School for Information Technology, The University of Jordan, Amman 11942, Jordan
dana.beidas@gmail.com

doi: 0.4156/ijiip.vol1.issue1.6

Abstract

With the emergence of multimedia databases, exact keyword search performed in traditional databases is not applicable due to the complex semantic nature of multimedia data. In this paper, Content Based Image Retrieval approach was introduced to solve this problem by providing metadata for multimedia databases based on their actual contents (features) rather than raw keywords description. The search is based on similarity matching rather than exact match because of the fact that images are rarely identical. The used images in the experiments were obtained from Grimace facial images dataset available from the University of Essex, England. Similarity between database objects (images) was calculated using Euclidean, City-block and Chi-square distance functions. The most attractive results of the conducted experiments were obtained using City-block and Euclidean distance functions. Image's features that can perform well when used individually were identified. Features that can perform well when combined with other features were also identified, in addition to excluding features that have limitations in distinguishing images such as image entropy value.

Keywords: *Database, Multimedia, Content based image retrieval, Distance function, Grimace.*

1. Introduction

Databases which are controlled collections of data that contain information related to a given entity are at the core of any information system [1]. Traditionally, databases are used to store raw data as Binary/ASCII files - ASCII files stand for American Standard Code for Information Interchange and sometimes are called plain text files - since it was composed primarily of textual and occasional graphics data. With rapid technological advances and high speed data communication networks; user expectations have consequently increased and the need for developing multimedia databases has become persistent need in the on-going applications. Multimedia databases are databases that contain images, audio, video and animation in addition to the textual data [2].

Because of having different data types involved; multimedia databases need to have all the functions of traditional databases, in addition to other new and enhanced functionalities and features for storing, processing, retrieving, transmitting and presenting a variety of media data types in a wide variety of formats. At the same time, they must adhere to numerous constraints that are absent in traditional databases such as real-time constraints for the transmission of media data [1].

In traditional databases, the search is usually based on a key given as part of the query and the search is either an exact, pattern matching or a range search. While in multimedia databases, the search is likely to be similarity search; the query result is a ranked list of similar data items rather than exact matches.

A lot of applications areas can utilise multimedia databases and its similarity distance search. Examples of such applications are hospitals and healthcare centres; by saving medical images of previous patients, and retrieving the most similar medical image(s) of previous patients to a given patient medical query image as well as their diagnoses and therapy to help diagnose the infection and get the required therapy by analogy [3]. Police investigation and security systems can benefit as well by saving suspected persons facial images to help investigators get the most similar facial image(s) of a given suspect from the set of suspected persons to help finding disguised thieves [4]. Commercial applications, entertainment systems, people in education, tourism agencies, and artists can have great value from multimedia databases by providing the capacity to store large size videos and images and presenting them in an integrated way [4] [5].

Multimedia databases tend to be very large and executing queries is computationally expensive. When having a huge multimedia database and a great number of comparisons, an efficient similarity distance is needed in place. Hence, in this paper Content Based Image Retrieval (CBIR) approach is introduced to solve this problem by providing metadata for multimedia databases based on their actual contents rather than raw keywords description. When a multimedia item like an image is stored into the database, it is analysed and an abstract representation of its features is saved into a feature vector or signature. Then, search is based on similarity matching of these features rather than exact match.

The proposed approach aims to provide the best combination of extracted features and similarity distance search function that retrieves the most similar results from a huge images' database, discards irrelevant results and works efficiently in terms of time and space by spending less similarity search running time as well as less memory space.

The remaining contents of this paper are organised into the following sections. Section 2 reviews the related work. Section 3 discusses the proposed method, while section 4 discusses the conducted experiments and reviews their results. Finally, conclusions are drawn and possibilities for future work are proposed in section 5.

2. Related work

Searching a multimedia database for an image and/or video is conducted using similarity distances rather than exact search. Similarity distance search is based on extracting features from the media data using CBIR instead of keyword metadata exact match and then indexing the media data using feature-based indexing.

CBIR is a computer vision application that processes information contained in an image and creates an abstraction of its content features such as colours, shapes, textures, or any other information that can be derived from the image. Thus, query operations deals with this abstraction rather than with the image itself, and results are a ranked list of the most similar database objects retrieved using a similarity distance function such as Euclidean distance rather than an exact match.

Generally, CBIR systems consist of three main modules: input module, query module and retrieval module classified under two phases: indexing and searching as originally presented by Mohamed et al. [6]. Under the indexing phase there are the input module and the query module. In this phase, input image and query image features are extracted such as colour, shape and texture and stored in a feature vector or signature. Then these features are stored along with the image itself in the database. While in the searching phase; there is the retrieval module, where query image's extracted features are compared with database images’ features to retrieve the most similar images using similarity distance measures [6].

Extracted image features are too restricted to describe images on a conceptual level, which may cause a semantic gap that is defined as the gap between the high-level user's query and the low-level extracted features. Kulkarni [7] has proposed a fuzzy approach to reduce the semantic gap by converting human's high-level queries to the low-level features processed by the computer. Kulkarni compared the performance of fuzzy image features and fuzzy similarity measure with various distance functions which are the Euclidean, City-block, Chi-square and histogram intersection. All these distance functions performed well when implemented with fuzzy image features except for the histogram intersection.

Another research conducted by Mohamed et al. [6] where the authors proposed an efficient feature extraction method that is efficient in terms of speed and space. By directly accessing image content and extracting the average of Discrete Cosine Transform (DCT) block coefficients. While Mohamed et al. used an efficient approach for normalising images and extracting features by DCT, but they only used the Euclidean distance function without comparing the feature extraction method on other distance functions such as City-block or Chi-square.

Tayyab et al. [8] have also used the DCT to extract features from images in addition to Discrete Wavelet Transform (DWT). They have compared the performance of 2D-DCT and 2D-DWT for artificial neural network-based face detection. In their experimental results Tayyab et al. claimed that the performance increases by increasing the number of images, and they concluded that for larger datasets 2D-DCT gives better results than 2D-DWT.

One of the key issues that should be determined is how to correctly extract and measure the similarity of content features. Hao [4] in his research has focused on how to efficiently extract the image features, mainly how to extract face regions. Hao's approach was based on selecting similar images from database by adopting three feature extraction methods: Moment Invariants, Principle

Component Analysis (PCA) and Wavelet Decomposition. One of the strengths of this approach was combining more than one strong feature to get the results; the same idea was adopted in our proposed approach but using different features than the ones selected by Hao.

As there is a huge amount of data in multimedia databases to be processed and analysed, the need to analyse this data efficiently with the least human interaction has been increased. This motivated Foschi et al. [9] to implement an effective methodology to automatically analyse images and detect their regions by means of features. They have discussed methods that are used to quickly extract features and to evaluate the efficiency of these features. Foschi et al. approach has an advantage of selecting different features such as colour, edge and texture, then checking which features are stronger than others in distinguishing images, and which features are not efficient when used individually.

Albanese et al. [10] have introduced an animate image similarity model that would be useful for processing CBIR in video segmentation and image retrieval. This model has the advantage of processing only small fraction of the information contained in the image, it does not process all the image contents, but instead it captures salient points and regions named as Focus of Attention (FOA). The procedure is a sort of inexact matching, which denotes animate matching summarised as selecting the homologous FOA points in the query image and computing the animate matching as the average of three consistencies: local spatial consistency, local temporal consistency and local visual consistency.

The proposed approach in this paper is distinguished from others previous studies in using various combinations of features (up to 14 different features for each image with total of 37 features for all images' approaches) and several distances functions (3 distance functions) to implement the goal of determining the most efficient combination in terms of obtaining accurate results from the database images that are similar to a given query image, requiring less time and space by spending less similarity search running time as well as less memory space.

3. Search in multimedia databases using similarity distance

This section presents details of the proposed method. The general outline of the methodology of the proposed system with the main steps is discussed, followed by information about the used images dataset and how features are extracted from the images. Finally, the used distance functions are described in details.

3.1. Research methodology

Four different approaches to perform feature extraction have been used in this study; which are extracting features from the original image then presenting them as a feature vector, extracting features from the image normalised after cropping, extracting features from the binary edge image, and extracting all these features and combining them as one feature vector.

For each one of these approaches, 14 different features were extracted; they are classified under the following five categories: (1) Histogram features, (2) Edge features, (3) Features extracted by the DCT, (4) Texture feature and (5) Statistical features. Different features have been selected to compare their performance and conclude which features are strong enough to be used individually for describing images, and which features can perform better when combined with other stronger features.

Histogram features include image histogram, features extracted by histogram equalization, invariant colour histogram and traditional colour histogram. Edge features include grey image Canny edge, grey image Sobel edge, black and while Canny edge and black and white Sobel edge. Features extracted by the DCT include the DCT of the whole matrix and the DCT of each 8X8 image blocks, texture feature is the entropy, and finally statistical features include: image range, standard deviation and mean.

After extracting the above features, each feature's value was normalised according to the existing values of this feature for that image approach and then features are combined in a feature vector.

In the first approach, all the above features of the original image were extracted and combined into one feature vector. In the second approach, original image was normalised by cropping then all the cropped image features were extracted and combined as one feature vector. In the third approach, the Canny edge image was generated from the original image and then all the edge image features were extracted and combined as one feature vector. Finally, in the fourth approach all these features specified in the previous three approaches were combined into one feature vector. The purpose is to study the best combination of features that would retrieve the best accuracy in terms of retrieved images while consuming minimum memory space.

The feature vectors form the previous approaches are used to judge to determine the most similar images to be retrieved from the database using a distance function, the image with the minimum distance value will be the most similar image to a given query image.

Three distance function have been used with each one of these approaches which are the Euclidean, City-block and the Chi-Square distance functions. Each one of these distance functions has its properties and advantages. There is 12 combination of feature vector and distance function.

After concluding the most efficient approach (combination), more focus have been given to this approach's features in order to identify what are the strongest features that can be used alone to get accurate results, and which features can perform well when combined with other features. In addition, inefficient features that get negative results always should be identified and excluded.

Since it is not enough to determine the efficiency of the approach and the extracted features based only on the retrieved accurate results, this study has focused on features that are efficient in terms of accurate results with the least required memory space and the fastest computation time.

3.2. Facial images' database

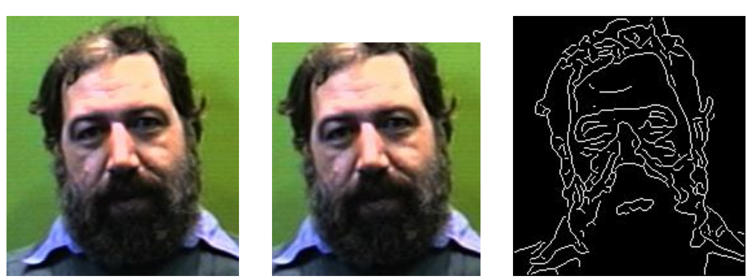
A standard facial images database obtained from the University of Essex, England has been selected; because they have high quality, coloured, includes several males and females facial images and stored in Joint Photographic Experts Group (JPEG) compressed format with variation in expression, head turn and position [11].

Four sets of images are available for selection, which are called: Faces94, Faces95, Faces96, and Grimace. Grimace facial database has been selected in order to focus the research on faces similarity under extreme facial expressions; as the Grimace database is distinguished from other databases as it is the most difficult database as it contains images having extreme variation of facial expressions, plain background, little variation in light and head scale, some translation in faces position in image with considerable variation in head turn, tilt and slant.

Grimace database contains a total of 360 images that are 24 bit colour JPEG with resolution of 180X200 pixels with portrait orientation. Images are for 18 individuals; both males and females and each with a sequence of 20 images representing different expressions, where the images were captured using a fixed camera.

Because of the fact that it is not permitted to publish or print the Grimace images themselves, an approval several images from the Faces94 database instead have been obtained taking into account that both Faces94 and Grimace databases have facial images with the same resolution of 180X200 pixels and both have JPEG format images with 24 bit colour. Faces94 is the simplest facial database for the purpose of presenting the extracted features. The permission was obtained through personal communication with the volunteer in the used images who is Professor Nadim Obeid who is working currently as a full professor and the dean of King Abdullah II School for Information Technology (KASIT) at the University of Jordan. Professor Obeid was working at the University of Essex when the database images were collected. The used images in the following examples are just to illustrate the operations that were performed on Grimace images.

Three representation approaches are used for each image used in the experiments; features are then extracted from each representation. These representations are illustrated in Figure 1.



(a) Original Image (b) Cropped Image (c) Canny Edge

Figure 1. The original image, cropped version of the image, Canny edges of the original image

The original image (named obeidn.1.JPEG) has a resolution of 180 by 200 pixels as shown in Figure 1 (a). The image is normalised by cropping operation, which would give more focus on the face as well decrease the impact of image's background. The image after cropping has a resolution of 160 by 180 pixels as shown in Figure 1 (b). The third image representation method used to extract features from is the Canny Edge of the original image as shown in Figure 1 (c).

3.3. Feature extraction methods

Five feature extractions methods are used in the proposed approach: histogram analysis, edge detection, DCT, texture analysis and statistical analysis. These features are extracted from the original image, cropped image and the Canny edge image except for the histogram analysis features which are not applicable to the Canny edge image. MATLAB software package (2007a, The MathWorks, Inc., Natick, MA, USA) image processing Toolbox was utilised to extract the needed features.

Image histogram and histogram equalization (imhist and histeq) are two of the main functions used for exploring histograms [12]. Image histogram displays a histogram of the image data, which can be either an intensity or binary. Imhist (I) function displays a histogram for the intensity image I above a greyscale colour bar. Histeq transforms the values in an intensity image to enhance the image's contrast by spreading the intensity values over the range of the image.

As these two functions are used for intensity images, the invariant colour histogram (invhist) developed by Domke and Aloimonos [13] was adopted to get the colour histogram for an image. This method can be used to get invariant colour histogram as well as traditional colour histogram. Invariant colour histograms are colour histograms that are invariant under any mapping of the surface. While the traditional colour histogram is a multidimensional histogram of the image's distribution of colour.

Edge detection methods take a two dimensional image as input and return a binary image of 1's where the edges are found and 0's elsewhere. Edges are places in the image corresponding to object boundaries were the image intensity changes rapidly. To find the edges, the facial images were converted from RGB (Red, Green, Blue chromatic images) to greyscale or to binary image, and then edge detection was executed.

Two edge detection methods have been applied: Canny and Sobel to the binary and greyscale images. Canny is distinguished from other methods in using two different thresholds to detect strong and weak edges, and including the weak edges in the output only if they are connected to strong edges. Thus, Canny method is more likely to detect true weak edges and less likely to be disturbed by noise.

When using Canny, it was found that converting images to greyscale instead of binary will produce better results in terms of edges precision. This can be explained by the fact that intensity images contain more details while black and white have just 0's and 1's values with no gradients.

DCT is a technique for converting a signal into elementary frequency components. DCT is used in JPEG image compression and it can be defined as:

$$f(u,v) = \frac{1}{\sqrt{2N}} c(i)c(j) \sum_{x=1}^{N} \sum_{y=1}^{N} f(x,y) \cos\left[\frac{(2x+1)i\pi}{2N}\right]$$

$$\cos\left[\frac{(2y+1)j\pi}{2N}\right] \quad , \quad i, j = 1 \ldots\ldots\ldots\ldots., N \qquad (1)$$

$$c(u) = \begin{Bmatrix} \frac{1}{\sqrt{2}} if u = 0 \\ 1 if u > 0 \end{Bmatrix}$$

Where

f (x, y) the element of the image f.

N is the size of the block that the DCT implemented

In this paper, two ways to generate features from the DCT have been implemented. Firstly, the image is transformed into a greyscale image then the average of DCT coefficient of the whole image is calculated. Secondly, the image is transformed into greyscale then it is divided into 8X8 blocks. After that, DCT is implemented on each of these blocks.

Texture feature of facial images was extracted using Entropy. Entropy is defined by Gonzalez et al. [14] as a statistical measure of randomness that can be used to characterise the texture of an input image. It returns a scalar value representing the entropy of greyscale image. Entropy is estimated in two stages. First, image histogram is estimated, and then the entropy is calculated. Entropy is defined by Gonzalez et al. as [14]:

$$-sum(p.*\log_2(p)) \tag{2}$$

Where p is the histogram counts returned from image histogram. Input image can be binary, greyscale or multidimensional image. If image is multidimensional, then entropy treats it as multidimensional greyscale and not as an RGB image.

Three statistical analysis features were extracted which are: (1) the standard deviation of the matrix elements,(2) the mean of the matrix elements and (3) the range of the matrix elements, returned value is a vector containing the range for each column.

For each image in Grimace database, features were extracted in the four approaches as following: (1) 14 features for the original image, (2) 14 features for the cropped image and (3) 9 features for the Canny edge image; since some features cannot be applied on the binary edge image (e.g.: image histogram equalization, invariant color histogram, traditional color histogram), (4) Combining all the previous features in one feature vector.

3.4. Distance functions

Three distance functions are used in the approach proposed in this paper: Euclidean, City-block and Chi-Square. Euclidean distance is the most common distance function which is an ordinary distance between two points that would be measured with a ruler. Euclidean distance function is appropriate when all the input attributes are numeric and have ranges of approximately equal width to avoid having a factor dominating the others. Euclidean distance is defined by Wilson and Martinez [15] as:

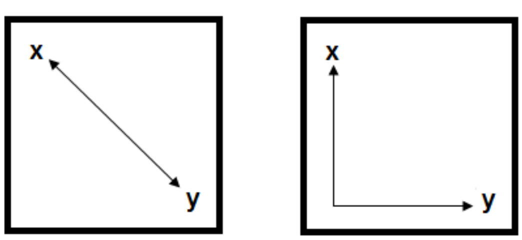
$$D(x,y) = \sqrt{\sum_{i=1}^{m}(x_i - y_i)^2} \tag{3}$$

Where x and y are the two input vectors, m is the number of input attributes, and xi and yi are the input values for input attribute i for the two instances under comparison.

While the Euclidean distance corresponds to the length of the shortest path between two points, the City-block (Manhattan) distance is the sum of distances along each dimension and requires less computation. As its name indicates; City-block distance function would be a suitable method to compute the distance between two points in a city where it is crowded with streets, traffic and fences and it is impossible to move straightforward from one point to another. Manhattan distance is defined as [15]:

$$D(x,y) = \sum_{i=1}^{m}|x_i - y_i| \tag{4}$$

The difference between Euclidean and City-block distance is illustrated in Figure 2.



(a) Euclidean distance (b) City-block distance

Figure 2. Euclidean distance vs. City-block distance

A variant of Euclidean distance and City-block is called the Chi-Square or weighted Euclidean distance is used to measure the distance between two points with a differential weighting of the dimensions of the space. It is defined by Wilson and Martinez [15] as:

$$D(x,y)=\sum_{i=1}^{m}\frac{1}{sum_i}\left(\frac{x_i}{size_x}-\frac{y_i}{size_y}\right)^2 \tag{5}$$

4. Experiments and results

In this section, the conducted experiments on the Grimace database (Section 3.2) are explained. Then the results of the experiments are shown along with an application of the proposed method using the image (obeidn.1) from the Faces94 database.

4.1. Experiments description

After extracting the features described in Section 3.3 for each image in the Grimace database, they are saved into a features' database. All image features are saved along with the image's name into a matrix structure inside MATLAB. Grimace database which contains 360 different images is divided into two sets: previous examples (training dataset) and unseen examples (testing set). The size of the training dataset was chosen to be 75% of the dataset (270 image) while the size of the testing dataset was chosen to be 25% of the dataset (90 images). The two datasets are disjoint; i.e. no image exists in both the training set and the testing set. Each image in the testing dataset will be used as a query image to get the most similar image from the training dataset. After the extraction of the features, each image feature is transformed into a 1 dimensional (linear) vector; so it could work with any feature regardless their data type.

Since the extracted features have substantially different ranges, features are normalised, after the normalisation all features are combined to compose a feature vector or signature of the image and are saved in a MATLAB file. Then, each image from the testing dataset is compared with all the previous examples in the training dataset images, and the distance between the test image and the training image (the distance between the two features' vectors) is calculated.

Images with the minimum distance values are the most similar images to a query image and the accuracy of each retrieved image is judged using an image ID as in Grimace database all images that belong to a specific person have the same ID.

After getting the most similar image(s) for each image in the testing dataset according to the used distance function, the efficiency of the implementation is determined by dividing the number of images in the testing dataset that were tested correctly (images for the same person were retrieved) by the number of all images in the testing dataset. The above experiment is repeated twelve times corresponding to the four combinations of features and the three used distance functions to decide which features are the best in distinguishing images and getting accurate results based on the query image.

4.2. Results and application

Experiment results showed that cropped image features and all image features approaches when combined are the best in terms of accurate results, since they achieved 100% correct results under the three used distances functions. Figure 3 summarises the results of the 12 different experiments that have been conducted for each of the 4 features extraction approaches and the 3 used distance functions.

Figure 3. Features performance under the three distance functions

It was found that City-block distance is a very efficient distance to be used with all features extraction approaches since it achieved 100% accurate results using all combinations of features. The same for Euclidean distance; since all the approaches performed perfectly with 100% accurate results under except for the edge image features which had a very good performance of 96.67%.

While in the Chi-square, it was noted that all features and cropped image features have 100% accurate results and original image features produce 98.89% accurate results. The lowest efficient combination was the Chi-square with edge image features extraction approach which has the value of 55.56%.

Although when all the features are used, 100% correct results were achieved, but the cropped image features are more efficient as small memory space and less computation time are required as they have less number of features. At the same time 100% accuracy is achieved with the cropped features. Further experiments have been conducted focusing more on these features, in order to identify the most descriptive features and thus being able to perform the similarity matching using the least number of features to save memory space and processing time.

Experiments on the 14 cropped features were conducted to study the performance of each feature when used individually in distinguishing images; this is repeated for each of the 3 used distance functions. From the results, it was found that some features are strong enough and can be used alone to produce 100% accurate results, while other features can perform well when combined with other features.

The following are strong features that produce 100% accurate results when used with Euclidean distance: image histogram, histogram equalization, image traditional colour histogram, image DCT, image 8X8 Blocks DCT and image range. Among these features, the most efficient features to be used with Euclidean distance in terms of the lowest memory requirement are the image traditional colour histogram, image histogram, and image range. The remaining features are strong but require more memory space.

Furthermore, other features have a very good performance under Euclidean distance in terms of distinguishing images and getting accurate results which are image invariant colour histogram, grey image Sobel edge, grey image Canny edge, BW image Sobel edge, BW image Canny edge by getting 98.89%, 98.89%, 90%, 83.33% and 82.22%, respectively. The image invariant colour histogram is the most efficient in terms of using the lowest memory space that is. The lowest efficient features used under Euclidean distance in terms of performance are image mean, image standard deviation and image entropy value by getting 60%, 55.56% and 5.56%, respectively.

When City-block distance was used, the following are strong features that get 100% accurate results: image histogram, histogram equalization, image invariant colour histogram, image traditional colour histogram, image DCT, image 8X8 blocks DCT, and image range. From these features, the most efficient in terms of the lowest required memory space are the image invariant colour histogram, image traditional colour histogram, image histogram, image range sequentially. The rest of features are strong but requires larger memory space.

Furthermore, other features had very good performance under City-block distance in terms of distinguishing images and getting accurate results which are grey image Sobel edge, grey image Canny edge, BW image Sobel edge, BW image Canny edge by getting 98.89%, 90%, 83.33%, 82.22%, respectively. The lowest efficient features used under city-block distance in terms of performance are image mean, image standard deviation and image entropy value by getting 60%, 55.56% and 5.56% respectively.

It was noted that no feature among the extracted 14 features can perform 100% accurately individually under the Chi-Square, but when combined some features they will produce 100% accurate results such as combining the following features (BW image Canny edge, image histogram, image traditional colour histogram, image standard deviation and image mean). The strongest features to be used with Chi-square distance function are image mean, image standard deviation, image traditional colour histogram, image DCT and image 8X8 blocks DCT by getting 60%, 55.56%, 55.56%, 54.44% and 51.11% accurate results and requiring the lowest memory space, respectively. But they are not strong enough to distinguish images based on these features individually. All the edge image features did not perform well in distinguishing images under Chi-square distance, in addition to the image histogram, histogram equalization, image invariant colour histogram and image range.

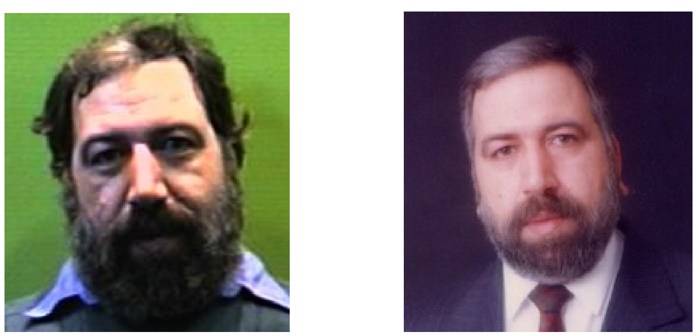
While conducting the above experiments, it was found that the used texture feature; Entropy is not a distinguishable feature for an image; this was confirmed when calculating distances between images based on entropy feature and produced just 5.56% accurate results. It will always get the same or near the value of 7.6439 for original images, 7.4919 for cropped images and for edge images it gave different values which were also not distinguishable for the images (person). Therefore, Entropy feature was excluded when computing similarity between images' features. It was also found that the following statistical features: image standard deviation and image mean have the same performance accuracy under the three used distances; it did not change at all.

All the features had the same performance when used with Euclidean and City-block except for the image invariant colour histogram. Image invariant colour histogram and image traditional colour histogram are efficient features in terms of distinguishing images and utilising the lowest memory space and are recommended to be used with Euclidean and City-block distance functions.

Moreover, Chi-square distance function is not a recommended distance function to be used with the selected features extracted from the Grimace database since it did not get efficient comparison results when compared with Euclidean and City-block distance functions.

The proposed method was applied on Faces94 facial database images in addition to Grimace database in order to get the most similar images to the example image obeidn.1 image. Faces94 database contains more than 3000 images for 153 individuals, including males and females. The query image from Faces94 database is obeidn.1 is shown in Figure 4 (a).

Another application of the proposed system was conducted based on a more recent query image of the same person Professor Nadim Obeid (obtained courtesy of Professor Obeid). The recent query image is presented in Figure 4 (b).

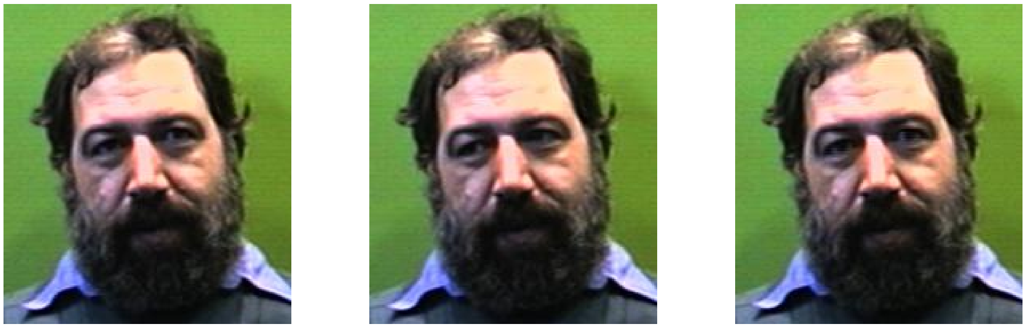


(a) Obeidn. 1 image from Faces94 database (b) Recent photo of Professor Obeid

Figure 4. Professor Obeid query images

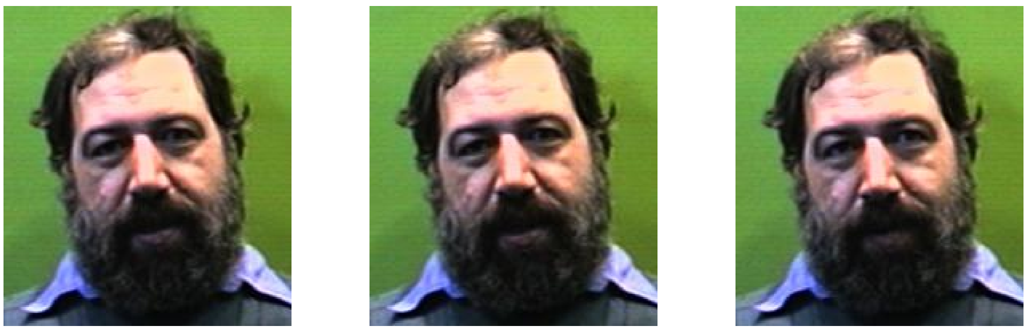
For each image, features were extracted from the cropped image approach, then these features are combined as a single feature vector for the image, then Euclidean distance was implemented to measure the similarity between the query and database images.

The proposed method was implemented on these two images to get the most similar three images, and results were 100% correct. The most similar images for Obeidn.1 query image were obeidn.3, obeidn.2 and obeidn.4 sequentially. The most similar images are shown in Figure 5 (a), (b), and (c). The most similar images for the recent photo for Professor Obeid were obeidn.20, obeidn.19 and obeidn.3 sequentially which are shown in Figure 6 (a), (b), and (c).



(a) obeidn.3 (b) obeidn.2 (c) obeidn.4

Figure 5. The most similar images to the Obeidn.1 query image



(a) obeidn.20 (b) obeidn.19 (c) obeidn.3

Figure 6. The most similar images to the recent photo query image

Obtaining accurate results even for a recent photo is an indication of the successfulness of the proposed approach in finding the most similar image(s) to a query image even after a while of time.

5. Conclusions and future work

In this paper, a method for images' retrieval from a multimedia database based on CBIR rather than keyword metadata was introduced. Images' retrieval was performed by similarity matching using three distance functions rather than exact match.

Four different approaches were used in this study which are: the original image's features, the cropped image's features, the Canny edge image's features and finally all these features combined as a single features vector.

It was found that cropped image features are the most efficient in terms of getting accurate results under all the used distance functions as well in terms of required memory since the image is normalised and cropped.

From the further experiments that have been conducted to identify the strongest features among all cropped image's features, it was found that there are efficient features that work perfectly with 100% accurate results such as image traditional colour histogram when used with Euclidean and City-block distances, while when using Chi-square distance function, it was found that features performed well when being combined with other features such as image histogram.

As a future work, it is proposed to use the cropped image features on other database images such as natural images to identify the strongest features that perform well in distinguishing images and getting accurate results when retrieving images based on a given query image. The similarity matching can be implemented by using various distance functions such as Euclidean and City-block or a combination of them.

More studies will be conducted on a multimedia database of videos, by splitting videos into frames, extracting the strong features of these frames and comparing the similarity between videos based on a query video using different distance functions.

The proposed approach can be used in search engines by allowing the user to enter a query image and/or video and retrieve the most similar images and/or videos stored in the multimedia database to this query object.

6. Acknowledgment

The authors wish to thank Professor.Nadim Obeid, the dean of King Abdullah II School for Information Technology (KASIT), The University of Jordan, Jordan for his help and permission to use his photos during the experiments of this paper.

7. References

[1] D. A. Adjeroh and K. C. Nwosu, “Multimedia Database Management Requirements and Issues”, IEEE Multimedia, vol. 4, no. 3, pp. 24-33, 1997.
[2] S. R. Subramanya, “Multimedia databases - Issues and Challenges”, IEEE Potentials, vol. 18, no. 5, pp. 16-18, 2000.
[3] E. Petrakis and C. Faloutsos, “Similarity Searching in Medical Image Databases”, IEEE Transactions on Knowledge and Data Engineering, v. 9 n. 3, p.435-447, 1997.
[4] S. Hao, “Image Features Extraction for Multimedia Database Content Description”, In Proceedings of the 5th IEEE/ACIS International Conference, pp.327-332, 2006.
[5] L. Dunckley, Multimedia Databases: An Object-Relational Approach, Addison-Wesley, USA, 2003.
[6] A. Mohamed, Y. Weng, J. Jiang and S. Ipson, “An efficient face image retrieval through DCT features”, In Proceedings of the 10th IASTED International Conference on Signal and Image Processing, pp.127-132, 2008.
[7] S. Kulkarni, “Image Retrieval Based on Fuzzy Mapping of Image Database and Fuzzy Similarity Distance”, In Proceedings of the IEEE/ACIS International Conference, pp.812-817, 2007.
[8] M. Tayyab, Z. M. F. and S. Ali, “Performance Comparison of 2D-Discrete Cosine Transform and 2D-Discrete Wavelet Transform for Neural Network-Based Face Detection”, In Proceedings of the of Soft Computing and Pattern Recognition, pp. 387-392, 2009.
[9] P. G. Foschi, D. Kolippakkam, H. Liu, and A. Mandvikar, “Feature extraction for image mining”, In Proceedings of the 8th International Workshop Multimedia Information. Systems, pp. 103-109, 2002.
[10] M. Albanese, A. Chianese, V. Moscato and A. Picariello, “Experience of Animate Similarity Concepts in Multimedia Database”, In Proceedings of the 22nd International Conference on Data Engineering Workshops, pp. 11-16, 2006.
[11] L. Spacek, Description of the Collection of Facial Images, http://cswww.essex.ac.uk/mv/allfaces /index.html, Computer Vision Science Research Projects, University of Essex, UK, 2007, Last visited: 30 August 2010.
[12] Y. Gonzalez and J. Rocha, “Some Practical Assignments in Computer Vision”, In Proceedings of the Workshop on Education and Practice in Artificial Vision, pp. 101-104, 2003.
[13] J. Domke and Y. Aloimonos, “Deformation and viewpoint invariant color histograms”, Proceedings of British Machine Vision Conference, pp. 509-518, 2006.
[14] R. Gonzalez, R. Woods and S. Eddins, Digital Image Processing Using MATLAB, (1st ed.), New Jersey, Prentice Hall, Chapter 11. 2003.
[15] D. R. Wilson and T. Martinez, “Improved Heterogeneous Distance Functions”, Journal of Artificial Intelligence Research, vol. 6, no. 1, pp. 1- 34, 1997.